AD-A052 714 ARINC RESEARCH CORP SANTA ANA CALIF F/G 5/1
DEVELOPMENT OF ON-LINE SOFTWARE PACKAGE FOR CALCULATING ACQUISI--ETC(U)
APR 77 P J ORTH F33657-77-D-0029
UNCLASSIFIED W77-1974-TN01 NL

| OF |
AD
A052 714

END
DATE
FILMED
5-78
DDC

DEVELOPMENT OF ON-LINE SOFTWARE PACKAGE FOR CALCULATING ACQUISITION COSTS

April 1977

Prepared for

AERONAUTICAL SYSTEMS DIVISION
Wright-Patterson AFB, Ohio

Under Contract F33657-77-D-0029-0004

Publication W77-1974-TN01

ARINC RESEARCH CORPORATION
P.O. Box 1375/Santa Ana, Calif.

(6)

DEVELOPMENT OF ON-LINE SOFTWARE PACKAGE FOR CALCULATING ACQUISITION COSTS.

(11) Apr[illegible]77 (12) 24p.

(9) Rept. for Mar-Apr 77,

Prepared for

AERONAUTICAL SYSTEMS DIVISION
Wright-Patterson AFB, Ohio

Under Contract F33657-77-D-0029-0004

(15)

Prepared by

(10) P.J. Orth

ARINC RESEARCH CORPORATION

CORPORATE HEADQUARTERS
2551 Riva Road
Annapolis, MD 21401

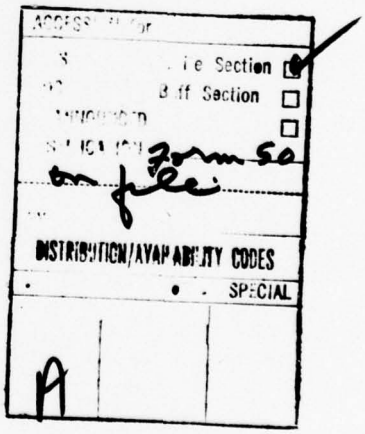
SANTA ANA BRANCH
1222 E. Normandy Place
Santa Ana, CA 92704

(14)

Publication W77-1974-TN01

400 711 LB

CONTENTS



1
INTRODUCTION AND SUMMARY

This report documents the activities of ARINC Research Corporation under Contract F33657-77-D-0029-0004 during March and April 1977, and delineates the planned future efforts.

Under the subject contract, ARINC Research assisted the ALCM Program Office by developing an on-line computer program that allows a large spectrum of potential users, including those not familiar with computer programming, to calculate acquisition costs for a wide range of scenarios. The program provides outputs in a generalized format as specified by each user. Users have available a large selection of routines that can be called into operation as needed, and which will operate on input data as specified by the user. The program is now operational at the computer facility at Wright-Patterson Air Force Base, Dayton, Ohio.

From the inception of this study, it was realized that the computer program developed would be a dynamic entity, subject to continuing modification to include additional methods of calculation and to adapt its requests (prompting statements) to the "knowledge" of the user population and their general familiarity with the program. Indeed, the program was created with the idea in mind of ease of enhancement and change.

As the project progressed, it became clear that all initial objectives would be attained — and considerably more. This enabled early implementation of certain parts of the program that provided insight into its general applicability and what could be attained with additional routines. Thus, although the program was created to calculate ALCM acquisition costs, it became obvious that it could equally well make calculations for any program.

For the above reason, a follow-on effort is in the process of being funded by the Financial Management Office of the RPV/ALSM System Project Office to provide program enhancement. Thus the computer program that now exists will shortly cease to exist, having been modified. This being the situation, it was decided that this report

should summarize the effort since the last progress report. The final report, to be issued after the follow-on effort, will address the work performed under both this and the follow-on activity.

2
MARCH AND APRIL EFFORT

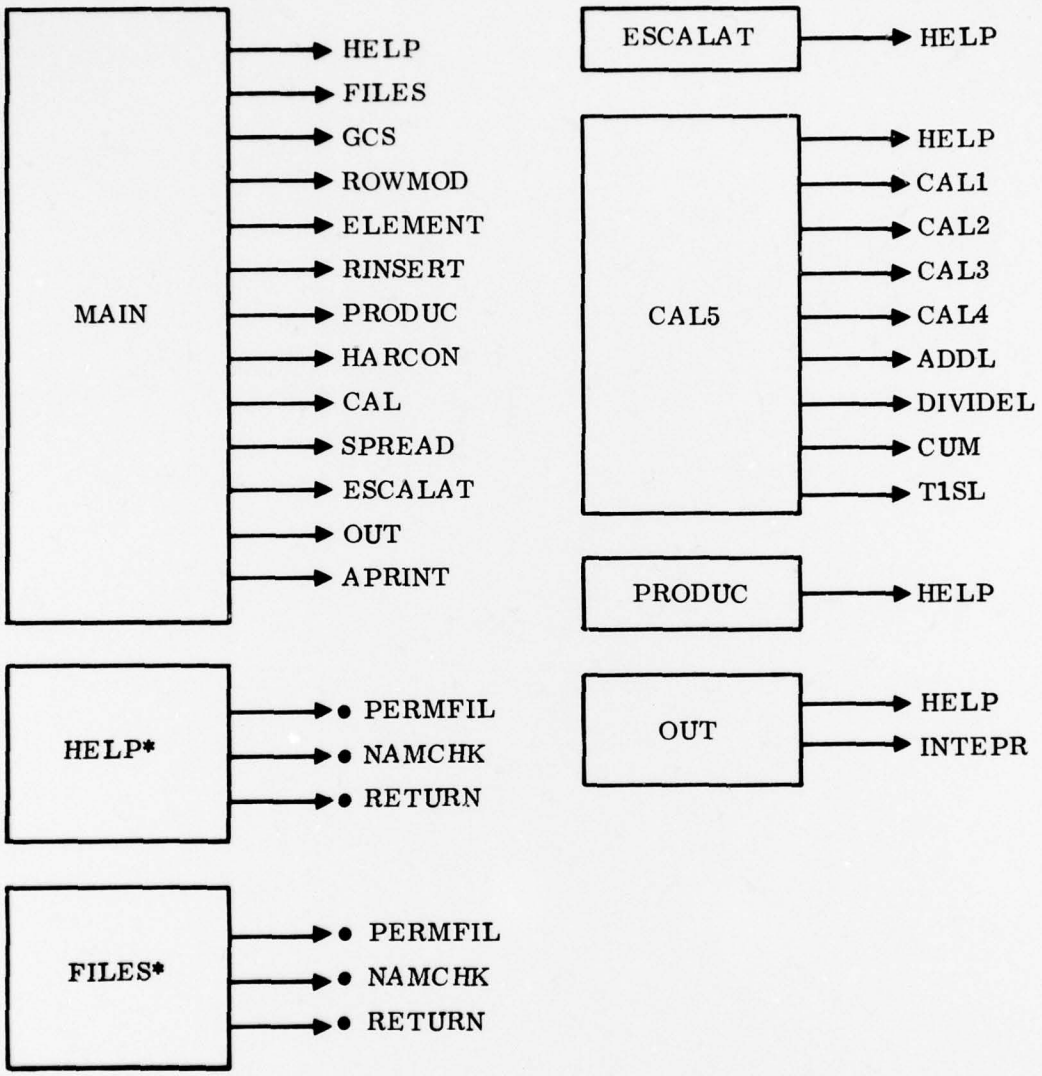
During the last two months of the contract period, 11 subroutines were added to the 14 developed during the first month. In addition, major changes were required in the MAIN routine, since it now provides interface for 11 more subroutines; and in subroutine PRODUC, which now enables simultaneous operation with 10 production schedules as opposed to only one previously. Minor changes were made in other subroutines to improve prompting and software reliability; and to enable interface, where required, with the new subroutines.

The new subroutines are described in Section 4.1, and the associated user's manual in Section 4.2.

3
COMPUTER PROGRAM OVERVIEW

Figure 1 provides an overview of the subject computer program, delineating the interfaces between routines. In general, such interfaces are dependent on an affirmative response by the user. In some instances, however, one routine calls another automatically, as indicated by "•" in the figures.

The delineation in Figure 1 includes the 14 routines developed during February 1977. The new routines developed during March and April are discussed in Section 4.



*These routines are called at the Wright-Patterson computer facility. PERMFIL and RETURN are system routines, and NAMCHK is a routine not yet added to the set of routines on the Control Data time-share network. On that network, system routine PFSUB is called.

Figure 1. Computer Program Overview
(Sheet 1 of 2)

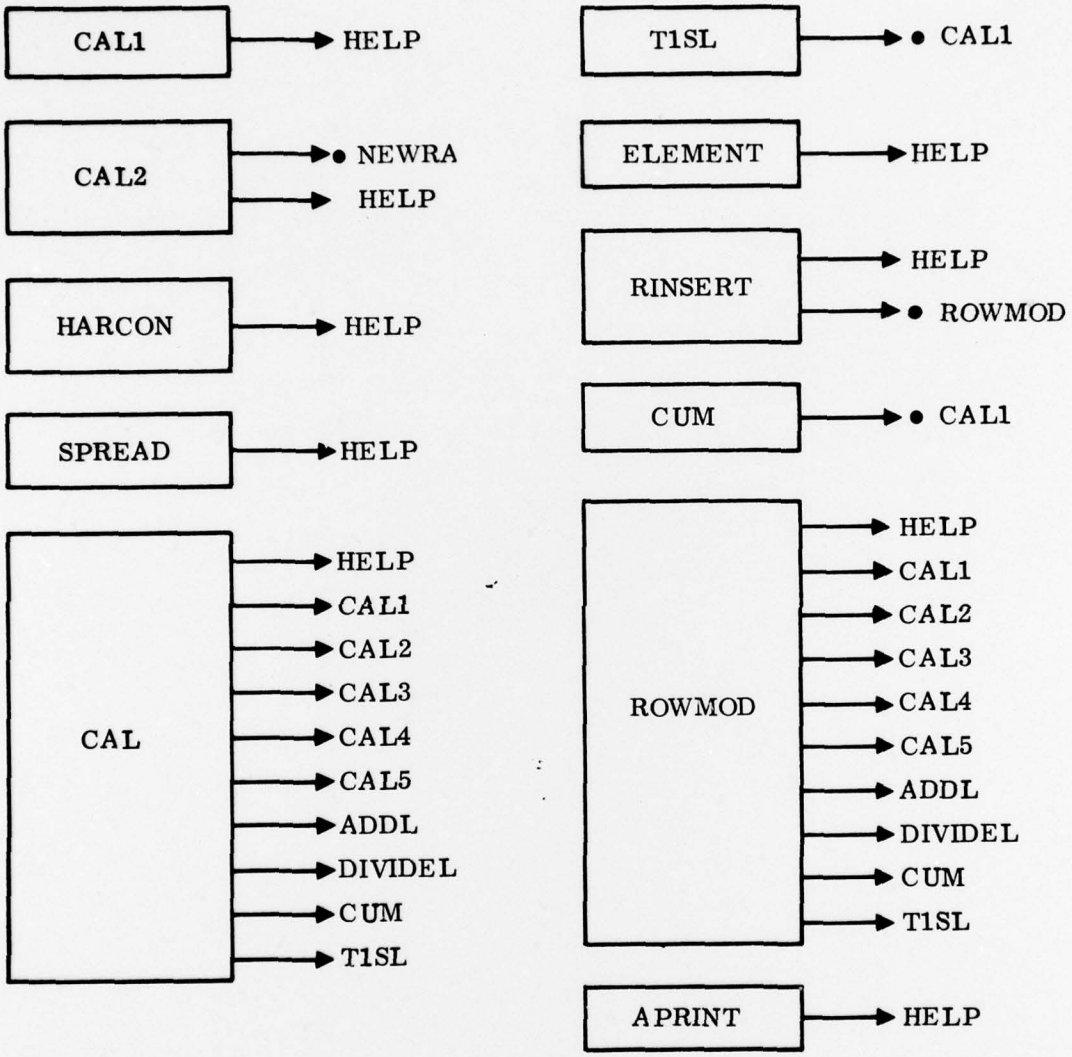


Figure 1. (Sheet 2 of 2)

4
NEW ROUTINES

This section presents a brief description of the newly developed acquisition-cost routines, and an abbreviated version of applicable sections of the User's Manual being prepared under this contract.

4.1 DESCRIPTION OF ROUTINES

4.1.1 Subroutine FILES(A, HEADW, PRODM, I19)

This subroutine enables data stored in files to be input to the cost, heading, and production schedule arrays during program operation. Additionally, information stored in these arrays can be placed in files during program operation.

4.1.2 Subroutine ADDL(A, NYEARS, COST)

This subroutine adds any number of specified rows from the cost array to form a new row. All elements of this new row are then multiplied by a specified constant.

4.1.3 Subroutine DIVIDEL(A, NYEARS, COST)

This subroutine enables the formulation of a row through the division of an existing row by a second existing row.

4.1.4 Subroutine INTEPR(INUM, NX, NROWS, A, HEADW)

This subroutine enables integer format output.

4.1.5 Subroutine APRINT(NYEARS, NROWS, A, HEADW, PRODM)

This subroutine outputs what is presently in the heading, cost, or production schedule array.

4.1.6 Subroutine ROWMOD(HEADW, A, PRODM, NYEARS, COST)

This subroutine enables the modification of a row in the heading, cost, or production schedule arrays. Row modification can be repeated as often as desired.

4.1.7 Subroutine CUM(COST, PRODM, NYEARS)

This subroutine determines first-unit cost, given the learning rate, the total cost of a sequence of items, and the first and last number in that sequence.

4.1.8 Subroutine RINSERT(A, NYEARS, NROWS, HEADW, PRODM)

This subroutine enables the insertion of a row in the heading and cost arrays. Insertion of rows can be repeated as often as desired.

4.1.9 Subroutine ELEMENT(A)

This subroutine enables an element of the cost array to be formed by specification, or by dividing a specified element of the cost array by another specified element.

4.1.10 Subroutine GCS(N)

This subroutine enables direct transfer from almost any location in the main routine to 15 designated locations in the main routine.

4.1.11 Subroutine T1SL(COST, PRODM, NYEARS)

Given two specific production sequences and the cost of each, this routine calculates the associated first-unit cost and learning rate.

4.2 USER'S MANUAL

This subsection contains a brief version of that section of the User's Manual relating to the new routines. The final version will be somewhat expanded, containing numerous examples. However, as will be discussed in Section 5, the present version of the program will undoubtedly be changed – both in the sequence of the requests and in the structuring of the requests themselves. Hence, it did not seem appropriate at this point to attempt to prepare a detailed version of the manual.

4.2.1 Subroutine FILES(A, HEADW, PRODM, I19)

This subroutine enables data stored in arrays, presently only the production schedule, cost, and heading arrays, to be saved under a unique name in files. During future runs, this routine enables these files to be used as default arrays. The requests posed by this routine, with associated explanation, are discussed below.

a. ENTER 1 IF YOU WISH TO SPECIFY, BY FILE, THE HEADING, COST AND/OR PRODUCTION ARRAYS, RESPECTIVELY, OTHERWISE ENTER 2. A TYPICAL RESPONSE WOULD BE 1, 2, 1 – The user is requested to specify those files to be used as arrays. The desire to use a particular array is delineated by entry of the numeral 1, a negative response by 2. Individual responses required are for the heading, cost, and production schedule arrays, in that order and separated by commas. Thus, for example, if it is desired to use only a cost array the response would be:

2, 1, 2

b. SPECIFY THE FILE TO BE READ INTO THE HEADING ARRAY – The user responds with the name under which the file to be used as the heading array is stored.

c. SPECIFY THE FILE TO BE READ INTO THE COST ARRAY – The user responds with the name under which the file to be used as the cost array is stored.

d. SPECIFY THE FILE TO BE READ INTO THE PRODUCTION ARRAY – The user responds with the name under which the file to be used as the production schedule array is stored.

e. FOR THE HEADING, COST AND/OR PRODUCTION ARRAYS, RESPECTIVELY, ENTER 1 TO SAVE ON FILES, OTHERWISE ENTER 2. A TYPICAL RESPONSE WOULD BE 1, 2, 1 – The user is requested to delineate which of the heading, cost, and production arrays are to be saved. Entry of the numeral 1 indicates the desire to save. Thus if only the cost array is to be saved, the response would be

2, 1, 2

f. INPUT THE NAME YOU CHOOSE TO GIVE THE FILE, STORING YOUR HEADING ARRAY – A UNIQUE FILE NAME NOT EXCEEDING 7 CHARACTERS – If the heading array is to be saved for future use, it must possess a unique name. The user is requested to input a name of his choice at this point. The name should begin with an alpha character and not exceed 7 characters.

g. INPUT THE NAME YOU CHOOSE TO GIVE THE FILE, STORING YOUR COST ARRAY – A UNIQUE FILE NAME NOT EXCEEDING 7 CHARACTERS – If the cost array is to be saved, it must possess a unique name. The user is requested to input a name of his choice at this point. The name should begin with an alpha character and not exceed 7 characters.

h. INPUT THE NAME YOU CHOOSE TO GIVE THE FILE, STORING YOUR PRODUCTION ARRAY – A UNIQUE FILE NAME NOT EXCEEDING 7 CHARACTERS – If the production schedule is to be saved, it must possess a unique name. The user is requested to input a name of his choice at this point. The name should begin with an alpha character and not exceed 7 characters.

4.2.2 Subroutine ADDL(A, NYEARS, COST)

This subroutine calculates a row of data by adding several specified rows, previously calculated, and multiplying this sum by a specified constant.

a. ENTER THE NUMBER OF ROWS TO BE ADDED AND THE FRACTION BY WHICH THE SUM OF THESE ROWS WILL BE MULTIPLIED – If 4 rows are to be added and multiplied by 1.35, the user responds

4, 1.35

b. ENTER THE ROWS TO BE ADDED – If the first, sixth, eight, and ninth rows are to be added the user responds

1, 6, 8, 9

4.2.3 Subroutine DIVIDEL(A, NYEARS, COST)

This subroutine calculates a row by dividing a previously calculated row by a second. Specifically, an element of the row, say in the second column, is calculated by dividing the element in the second column of the "numerator row" by the element in the second column of the "denominator row".

ENTER THE ROW OF THE NUMERATOR AND THEN THE ROW OF THE DENOMINATOR – If the third row is to be divided by the second row the user responds

3, 2

4.2.4 Subroutine INTEPR (INUM, NX, NROWS, A, HEADW)

This subroutine provides an integer output. No requests are made.

4.2.5 Subroutine APRINT (NYEARS, NROWS, A, HEADW, PRODM)

Upon request, this subroutine outputs the present contents of the heading, cost, or production schedule arrays.

a. TO OUTPUT THE COST ARRAY ENTER 1, THE HEADING ARRAY 2 OR THE PRODUCTION SCHEDULE ARRAY 3 – The response to this request is evident. If a printout of the heading array is desired, a 2 is entered.

b. IF ANOTHER ARRAY IS TO BE OUTPUT ENTER 1, OTHERWISE 2 – If the user wishes another array displayed, a 1 is entered.

c. ENTER THE NUMBER OF PRODUCTION SCHEDULES – The user responds with the number of different production schedules.

4.2.6 Subroutine ROWMOD (HEADW, A, PRODM, NYEARS, COST)

This subroutine enables the changing of values in a row of an array without having prompting requests made for every row in that array. The routine is applicable to the heading, cost, and production schedule arrays.

a. TO CHANGE A ROW IN THE HEADING, COST OR PRODUCTION ARRAY, ENTER 1, 2, OR 3, RESPECTIVELY – The user now delineates the array to be changed. Later, the user will be given the opportunity to specify a second array, or to change a second row in the specified array.

b. ENTER THE ROW NUMBER – The user specifies the row to be changed.

c. ENTER THE HEADING FOR ROW (row number specified in I3 format) – The user specifies the row heading for the row of interest.

d. TO MODIFY ANOTHER ROW ENTER 1, OTHERWISE 2 – If the user wishes to modify another row in the heading, cost, or production schedule arrays, a 1 is entered.

e. ENTER THE ROW NUMBER – The user specifies the row to be changed.

f. ENTER THE INDEX FOR THE METHOD OF CALCULATION – The user now delineates the way the row is to be calculated. See subroutine CAL for a listing of the methods available, together with their corresponding index.

g. ENTER THE ROW NUMBER – The user responds with the number of the row to be changed.

4.2.7 Subroutine CUM (COST, PRODM, NYEARS)

This subroutine determines first unit cost, given the learning rate, the total cost of a sequence of items, and the first and last number of that response. Subroutine CAL1 is then used to calculate the costs.

ENTER THE CUMULATIVE COST, THE FIRST AND LAST UNITS, AND THE LEARNING RATE – The user responds with the cumulative cost, the first and last units obtained in the sequence for which the cost was incurred, and the learning rate associated with the effort that produced the units and the cost. For example, if the total cost of the first and second units was $1,900,000 and the learning rate in producing those units was 0.9, then the user would respond

1900000., 1, 2, .9

The user then would receive the response (for this particular set of input data):

THE FIRST UNIT COST IS .10000E+07

The user does not respond to this output. The request/response sequence for Subroutine CAL1 is immediately initiated by the computer.

4.2.8 Subroutine RINSERT (A, NYEARS, NROWS, HEADW, PRODM)

This subroutine enables insertion of rows in the heading and/or cost arrays. Repeated insertions are permitted.

a. ENTER THE NUMBER OF THE TWO ROWS, SMALLEST FIRST, BETWEEN WHICH THE ROW WILL BE INSERTED – The user responds by delineating the location of the row to be inserted. Thus, to insert a row between rows 3 and 4 the user responds

3, 4

At this point the request/response sequence for ROWMOD is commenced. Preparatory to this sequence, however, the computer provides the following instructions for the use of ROWMOD. (As per the 3, 4 input in a, above. Also assume that the array contained 10 rows before the row insertion, so that there will be 11 rows after the insertion.)

> THERE ARE NOW 11 COST ELEMENTS. (ROWS) YOU ARE NOW GOING TO USE A ROUTINE WHICH MODIFIES ROWS. YOU WILL WISH TO MODIFY ROW 4 IN THE HEADING AND COST ARRAYS

The user does not respond to these instructions.

b. IF YOU WISH TO INSERT ANOTHER ROW ENTER 1, OTHERWISE 2 – The user is provided with the opportunity to insert another row.

4.2.9 Subroutine ELEMENT(A)

This subroutine enables the user to specify a particular element in his cost array, by providing the number to be stored; or indirectly, by specifying two other elements of the arrays, the quotient of which is to be used as the stored value.

a. IF THE ELEMENT IS TO BE FORMED BY SPECIFICATION ENTER 1, IF BY DIVISION ENTER 2 – The user indicates how the element of the array will be formed.

b. Depending on the answer to the above question, the request/response sequence is as follows:

1) If a response of 2 is made, then the following sequence occurs.

a. ENTER THE ROW AND COLUMN OF THE NUMERATOR ELEMENT, FOLLOWED BY THE ROW AND COLUMN OF THE DENOMINATOR ELEMENT, FOLLOWED BY THE ROW AND COLUMN OF THE ELEMENT TO BE CALCULATED. FOR EXAMPLE 10, 7, 22, 5, 24, 8 – Consider the example provided in the above request. Assume that the number 100 is stored in the 10th row, 7th column of the cost array, and that the number 4 is stored in the 22nd row, 5th column. Then the number 25 (100/4) would be stored in the 24th row, 8th column of the cost array.

b. IF ANOTHER ELEMENT IS TO BE OBTAINED ENTER 1, OTHERWISE 2 – The user delineates whether another element is to be changed.

2) If a response of 1 is made to the first question, a, then the following sequence occurs.

a. SPECIFY THE VALUE OF THE ELEMENT, ITS ROW AND COLUMN – The user specifies the value to be stored and the row and column of that location. The request delineated in item b (4.2.9b1b) is now made.

4.2.10 Subroutine GCS(N),

This subroutine enables immediate transfer from any of 15 locations in the MAIN routine to any of the other 14 locations. That the user is at one of the 15 locations can be ascertained by noting that the request at that location ended in a "#". Specification of the new location is made by entering the number (1000 + statement number of the new location). These statement numbers will be delineated in the following text.

The following is output if the user responds to a request ending in "#" with the value 2000.

THE FOLLOWING IS APPLICABLE TO THE MAIN ROUTINE ONLY.
IF IN RESPONSE TO AN INTEGER REQUEST 1000 PLUS ONE OF CERTAIN SPECIFIED STATEMENT NUMBERS IS INPUT THEN THE USER IS SENT DIRECTLY TO THAT STATEMENT NUMBER. THUS, FOR EXAMPLE, A RESPONSE OF 1088 WOULD SEND THE USER TO STATEMENT 88 WHICH REQUESTS THE NUMBER OF YEARS OF INTEREST. THIS IS APPLICABLE FOR THE FOLLOWING STATEMENT NUMBERS.

STATEMENT NUMBER	REQUESTS
88	YEARS SPECIFICATION
400	ROWS SPECIFICATION
401	INPUT FILES
402	ROW MODIFICATION
403	ELEMENT MODIFICATION
404	ROW INSERTION
405	PRODUCTION SCHEDULE
406	HEADING ARRAY SPECIFICATION
77	CALCULATIONS
407	SPREADING THE DATA
408	ALLOWING FOR INFLATION

677	OUTPUT
409	ARRAY CHECK
410	STORE FILES
411	TERMINATE

4.2.11 Subroutine T1SL(COST, PRODM, NYEARS)

Given two specific production sequences and the cost of each, this subroutine calculates the associated first-unit cost and learning rate. Subroutine CAL1 is then called.

> ENTER THE FIRST AND LAST UNITS OF THE FIRST GROUP, THEN THE SECOND GROUP, FOLLOWED BY THE UNIT COST AND PRODUCTION NUMBER FOR THE FIRST GROUP, THEN THE SECOND – The user is requested to specify the two sequences by inputting first and last unit numbers for each group. Thus if one group was composed of items 17 through 39, and the second group 54 through 57, the user would input
>
> 17, 39, 54, 57
>
> This would be followed by average unit cost and the number of units, or by the total cost for those units and the number 1. Thus, if the total cost for the first group of units is 24 and the average unit cost for the second group is 0.9, then the complete input would be
>
> 17, 39, 54, 57, 24., 1, .9, 4
>
> where the last number, 4, represents the number of units in the second group.

5
FUTURE EFFORTS

Four general areas of program enhancement are envisioned, as discussed in the following paragraphs.

5.1 INPUT RETENTION/FILE INPUT

It is planned that program input data be automatically stored. The file thus created would not only preserve a record of the input, but enable sensitivity studies to be conveniently conducted through minor modifications to the file and then its subsequent use as the data input file.

5.2 SYSTEM ENHANCEMENT AND USER SUGGESTIONS

The on-line program as it presently exists constitutes a valuable analytical tool. Being on-line at the Wright-Patterson computer facility, it is readily available to that population of users for whom it was created. Their suggestions regarding the detail of prompting requests, program enhancement through additional routines, and software reliability will provide valuable assistance in enhancing the utility of the program during the follow-on effort.

5.3 CORE LIMITATION

The complete program, with all its subroutines, presently exceeds the allowable core. For present purposes, two of the less frequently used routines, SPREAD and INTEPR, have been omitted so the program could become operational. There are several possible solutions to this problem, all fairly easy to implement but having varying impact on the user population. Thus, care must be taken in choosing the means of resolving the core-limitation problem. It is planned that this decision will be made early in the follow-on effort so that it can be evaluated by the users.

5.4 ADDITIONAL ROUTINES

Some suggestions have been made concerning the need for new routines and modification of present ones. The suggestions include the following:

a. Column Insertion. Analysts are presently required to include a column in their cost array for costs incurred in prior years. This step would be best handled by adding the capability to the program for inserting a column in the cost array. This program modification will be accomplished.

b. Production Schedule Prompting. The program can now make calculations using any of 10 production schedules. Before each calculation, the user is prompted to input an index for the production schedule to be used. When only one production schedule exists, this is obviously a waste of the user's time and the request should be suppressed. Such a feature will be incorporated in the program.

c. Variable Level Prompting. It has been suggested that there be two levels of prompting: one for individuals familiar with the program, and a second for those not familiar. Implementation of this feature would, of course, take up some valuable core and might require considerable development time. This suggestion will be evaluated.

AD A052 714

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER W77-1974-TN01	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DEVELOPMENT OF ON-LINE SOFTWARE PACKAGE FOR CALCULATING ACQUISITION COSTS		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER W77-1974-TN01
7. AUTHOR(s) P.J. Orth		8. CONTRACT OR GRANT NUMBER(s) F33657-77-D-0029-0004
9. PERFORMING ORGANIZATION NAME AND ADDRESS ARINC Research Corp. 1222 E. Normandy Place Santa Ana, CA 92704		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AERONAUTICAL SYSTEMS DIVISION Wright-Patterson AFB, Ohio		12. REPORT DATE April 1977 13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) AERONAUTICAL SYSTEMS DIVISION Wright-Patterson AFB, Ohis		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

UNCLASSIFIED/UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)